A Practical Failure Prediction with Location and Lead Time for Blue Gene/P

Ziming Zheng, Zhiling Lan,
Illinois Institute of Technology

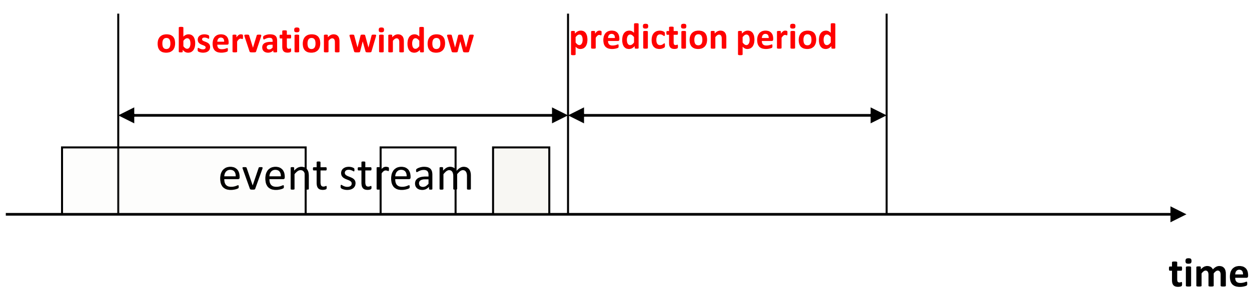
Rinku Gupta, Susan Coghlan, Peter Beckman
Argonne National Laboratory

Outline

- Motivations
- Background: Blue Gene/P
- Key contributions:
  - Refining Prediction Metrics
  - GA-based Prediction Method
- Experiments
- Conclusions

Existing Failure Prediction

- To learn failure patterns based on correlations between past events and fatal events
  - Examples: association rule, decision trees, Bayesian networks, support vector machines, …
  - They examine the events occurring during *observation window* and predict whether a fatal event will occur in *prediction period*



Issue #1 – No Location Information

- HEC systems are composed of thousands or more components
- *Location* is critical
  - Narrow down the potential problematic components
  - Take appropriate actions on failure-prone components, e.g., process migration and/or checkpointing
- Example: on Blue Gene/P, most of failures were reported at a single midplane or rack
  - A system-wide CKP (80 midplanes) may take up to 1,500 seconds, whereas a midplace- or rack-level CKP may only take ~120 seconds

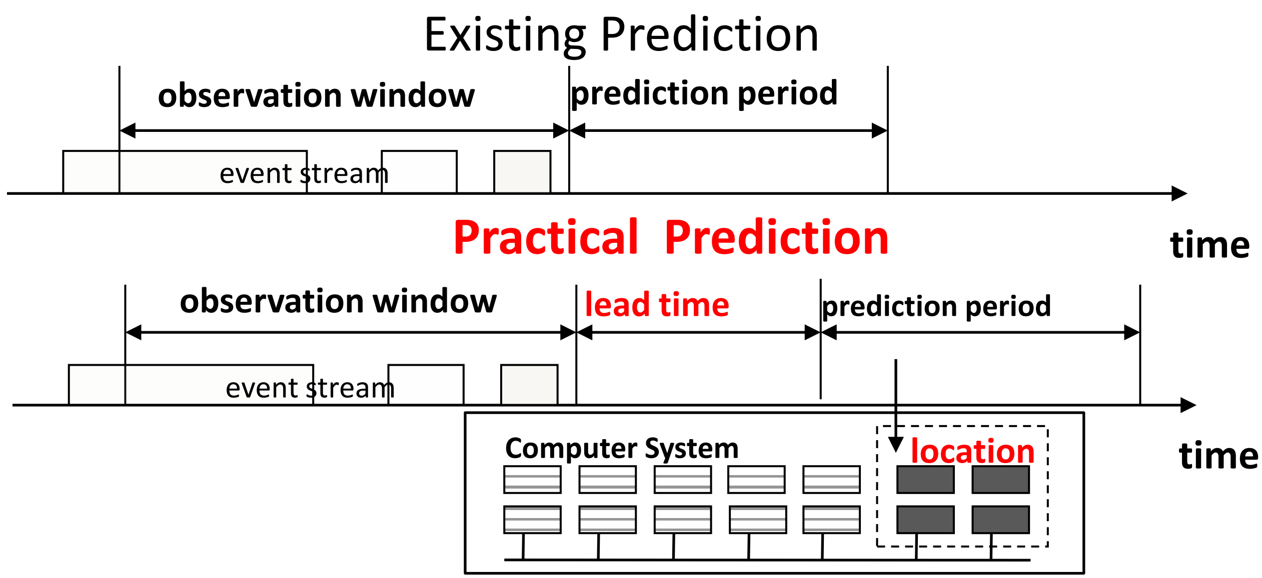
Issue #2 - Insufficient Lead Time

- *Lead time =* the time interval preceding the time of failure occurrence
- From practical usability perspective, lead time should be long enough to perform a fault tolerant action

- How to choose an appropriate lead time?
  - Predictions with high accuracy but short lead time may be useless in practice
  - A long lead time tends to reduce prediction accuracy

Our Contributions

1. Refine the traditional prediction metrics like precision and recall
2. Present a genetic algorithm based method for practical use on BG/P

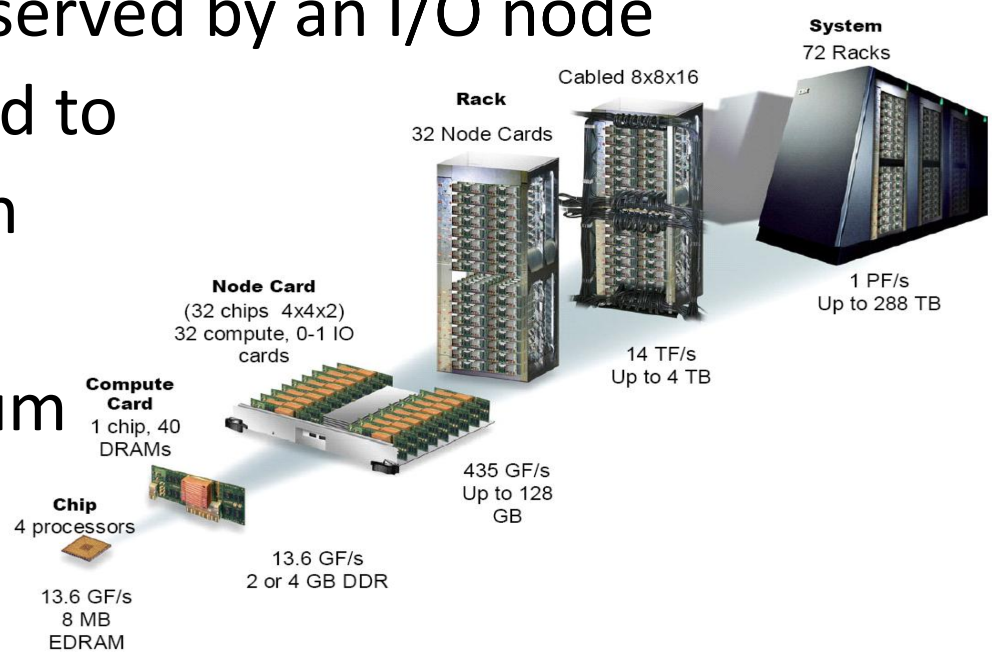


Outline

- Motivations
- Background: Blue Gene/P
- Key contributions:
  - Refining Prediction Metrics
  - GA-based Prediction Method
- Experiments
- Conclusions

Intrepid: Blue Gene/P system at ANL

- 40 racks/80 midplanes, 40,960 quad-core nodes
- No. 9 in the latest TOP500 list (June. 2010)
- 3D Torus-based network for compute nodes
- 64 compute nodes are served by an I/O node
- I/O nodes are connected to 136 file servers through a 10-Gigabit Ethernet
- Midplane is the minimum unit for job scheduling



Outline

- Motivations
- Background: Blue Gene/P
- Key contributions:
  - Refining Prediction Metrics
  - GA-based Prediction Method
- Experiments
- Conclusions

Prediction Metrics

- Precision and recall are two widely-used metrics to measure prediction accuracy.

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

	Actual Data		
Predicted Result		Fatal	Non-Fatal
	Positive	TP	FN
	Negative	FP	TN

- Location and lead time: complicating the defining of these metrics
  – Correct prediction of failure occurrence, but wrong location: FP and FN
  – System-wide prediction, midplane level failure: FP
  – Insufficient lead time: FN

Refining Metrics

- We refine the term of TP, FN, and FP
  - True Positive TP
    - Correct location & lead time > threshold
  - False Negative FN
    - No warning
    - Lead time < threshold
    - Wrong location information
  - False Positive FP
    - Warning on failure-free location
    - Wrong location information
- As a result, we refine *precision* and *recall* with the consideration of location and lead time

Prediction Rules

- Use a set of non-fatal events to predict fatal events *f*

$$< e_1, e_2, \cdots, e_k > \longrightarrow f$$

- Lead time = $\min(T^f - T^{e_i})$
- Location information
  - Choose one non-fatal event with the same location of fatal event
  - Three levels of location information: midplane, rack, or entire system

An example of a Prediction Rule

Non-fatal events

$< DGEMM\_MISCOMPARE, \_bgp\_err\_ddr\_single\_symbol\_error,$
$DGEMM\_SYNC\_NODES\_TIMEOUT > \longrightarrow$
$\_bgp\_err\_dma\_rec\_counter\_not\_enabled$

Fatal event

lead time: 325 seconds
location: the rack of $\_bgp\_err\_ddr\_single\_symbol\_error$

Rule Generation

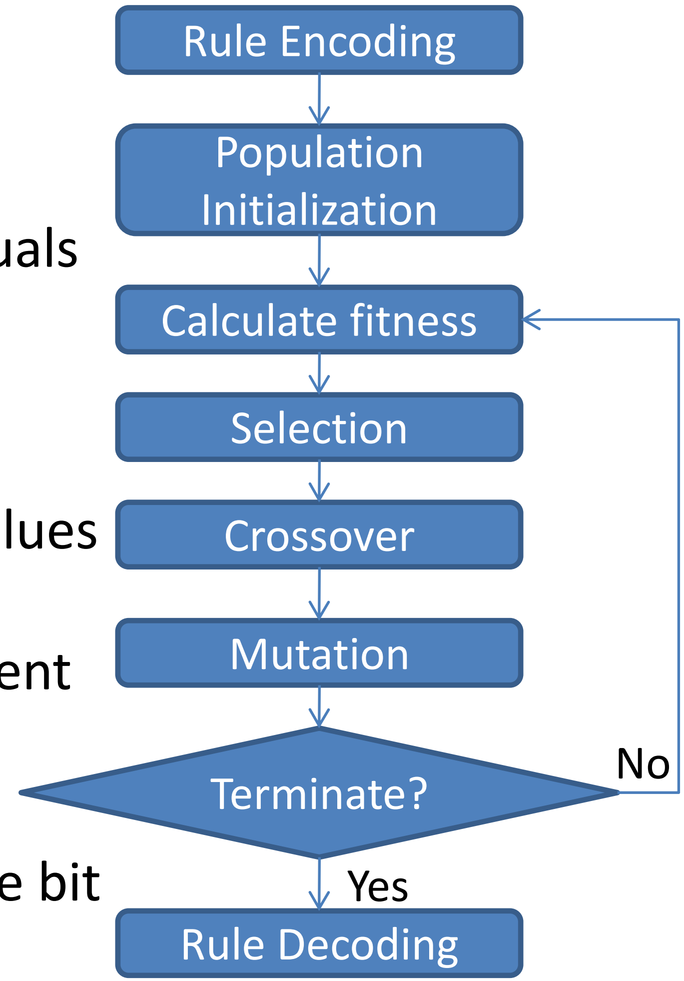
- Genetic algorithm based rule generation
  - GA is a widely used search technique for optimization problems
  - Various interacting parts in fitness function to address accuracy, location and lead time together
  - GA converges rapidly with a high probability to the rules with optimal or suboptimal accuracy

Rule Generation

- Michigan encoding
  - Transform rules to genetic individuals
- Initialize Population
  - Encoded random rules & elite individuals
- Fitness function

$$fitness = (w_1 \cdot recall + w_2 \cdot precision) \cdot W_{lead}$$

- Selection
  - Choose individuals based on fitness values
- Crossover
  - Copy some bits from two selected parent to breed new individuals
- Mutation
  - Make small random changes to a single bit in a genetic sequence



Outline

- Motivations
- Background: Blue Gene/P
- Key contributions:
  - Refining Prediction Metrics
  - GA-based Prediction Method
- Experiments
- Conclusions

Experimental Setting

- Evaluate our GA-based method by means of a real RAS log and a job log from Intrepid

Log Name	Days	Start Date	End Date	Log Size	No. of Records
RAS	81	2008-03-11	2008-05-31	3.5 GB	2715668
Job	31	2008-05-01	2008-05-31	4.5 MB	14108

- RAS log:
  - First being preprocessed using our method presented in DSN’09
  - Separated into two parts: the first 50 days as the training set and the rest of 31 days as the testing set
- Job log:
  - Used to examine the impact of failure prediction result on fault management

Z. Zheng, Z. Lan, B-H. Park, and A. Geist, "System Log Pre-processing to Improve Failure Prediction," Proc. of DSN'09, 2009.

RAS Events

- Event attributes:
  - Component: software component detecting and reporting the event
  - Severity: DEBUG, TRACE, INFO, WARNING,ERROR, or FATAL.
  - Errcode :fine-grained event type information.
  - Event Time: the time stamp
  - Location: the source of the event
  - Message: gives a brief description of the event

Rec ID	MSGID	COMPONENT	ERRCODE	SEVERITY	EVENT TIME	LOCATION	MESSAGE
13718190	CARD 0411	CARD	DetectedClock CardErrors	FATAL	2008-04-14-15.08.12.285324	R00-M0-N4-C9-U11	An error(s) was detected by the Clock card: Error=Loss of reference input

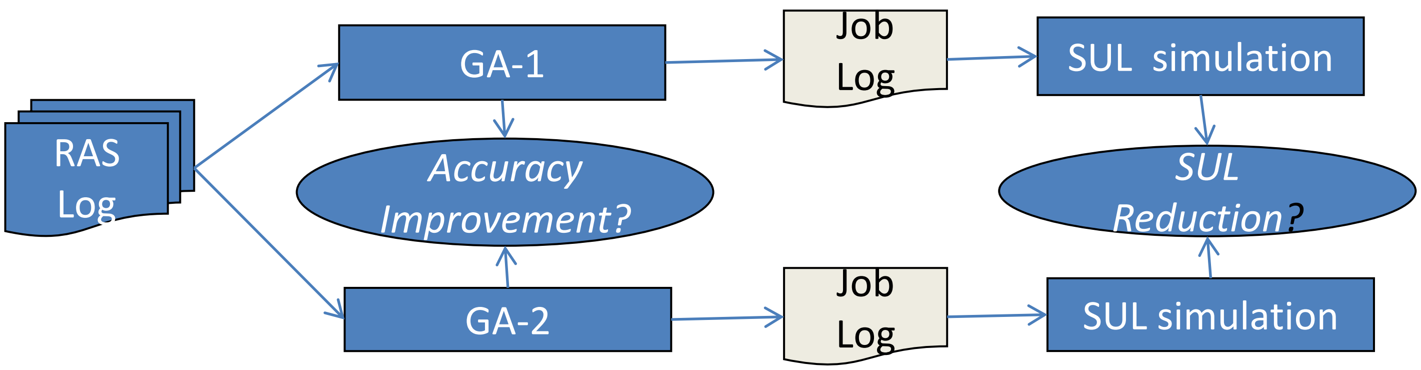
Job Events

- Job log attributes:
  - Queuing Time : Time when the job is added in the waiting queue.
  - Starting Time: Time when the job starts to run
  - End Time: Time when the job is finished or interrupted
  - Location : Execution units. Minimum unit is one midplane.

Job ID	Job Name	Execution File	Queuing Time	Starting Time	End Time	Location
8935	N.A.	N.A.	1209614949.07	1209618043.1	1209621636.96	R10-R11

Experimental Goal

- We compare two prediction methods:
  - *GA-1:* our GA-based method considering location and lead time
  - *GA-2:* a standard GA method without considering location and lead time
- Two goals: (1) examining prediction accuracy & (2) examining their impact on service unit loss

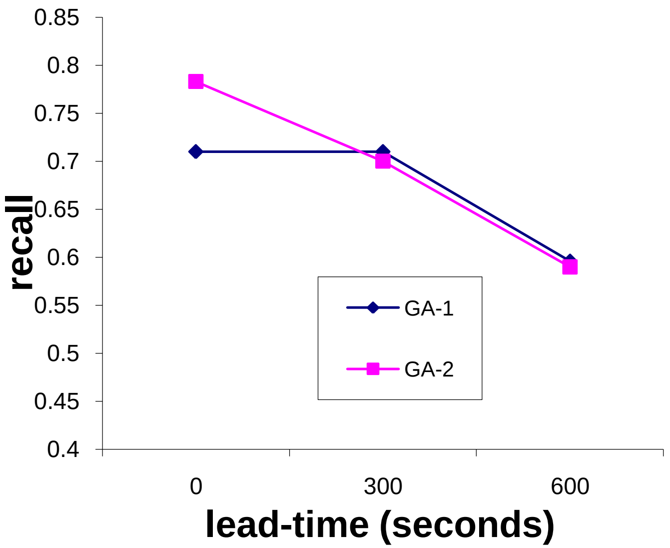


Results

- GA-1
  - Set the lower bound of lead time at 120 seconds to train
  - 10 rules provide midplane-level location
  - 7 rules provide rack-level location
  - 20 rules without location information

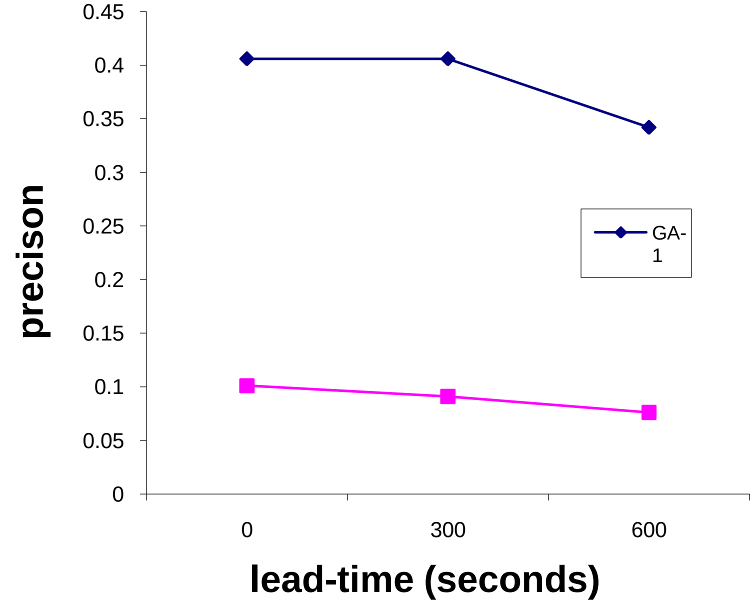
- GA-2
  - 41 rules without location and lead time information

Prediction Accuracy: Recall



- Recall decrease with a growing lead time
  - More precursor events cannot be used for prediction.
- GA-2 provides better recall when lead time is 0
  - GA-2 only provides prediction on system-level
  - GA-1 on midplane- and rack-level predications introduce FN.
- GA-1 outperforms GA-2 as lead time increases
  - GA-2 is prone to rely on events immediately preceding fatal events
  - GA-1 explicitly incorporates lead time in its fitness function

Prediction Accuracy: Precision

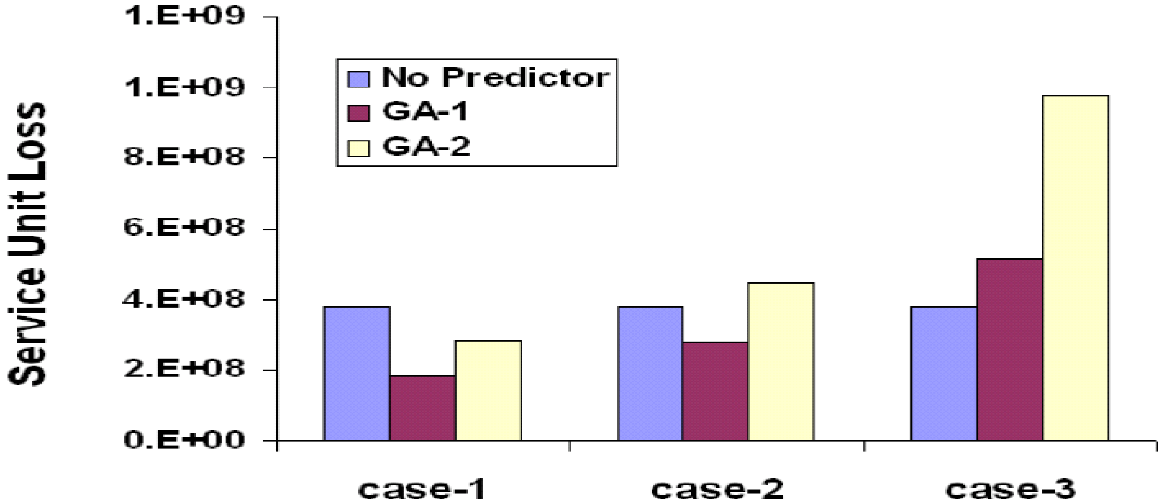


- GA-2 can only achieve about 0.1 on precision
  - 12 false alarms at system level = 12*80 false positives
- GA-1 can provide up to four times improvement
  - 5 false alarms at the system level, 7 at the midplane-level, 3 at the rack-level= 5*80+7+3*2 false positives

Impact on Fault Management

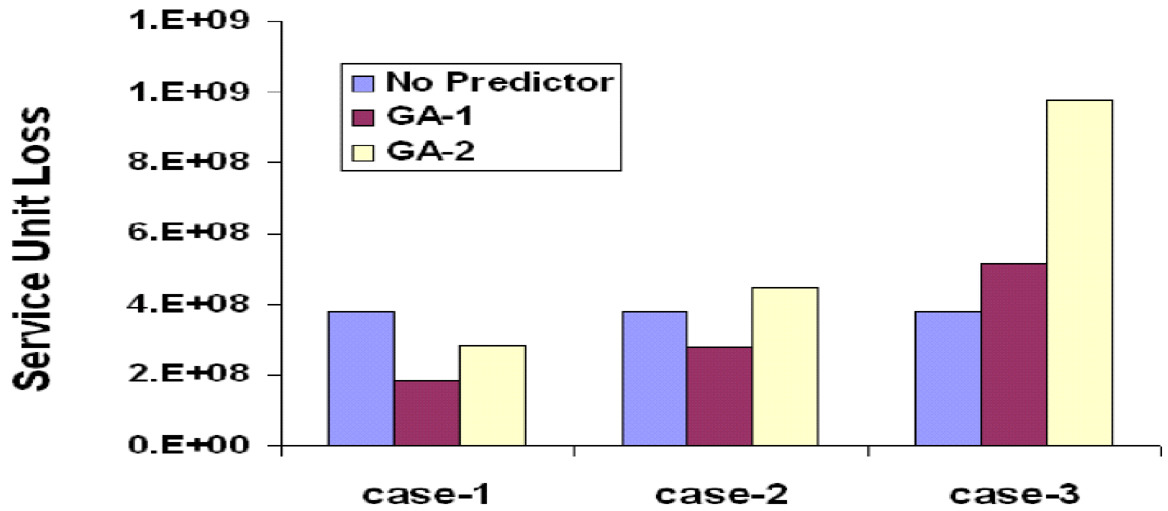
- Service unit loss *:* product of wasted wall clock hours and number of CPUs.
- SUL is traced out under three situations
  - Prediction miss leads to a job termination
  - Lead time is insufficient to conduct a checkpointing
  - System stops the job to issue a useless checkpointing due to false alarm
- Checkpointing overhead is estimated based on image size and available bandwidth
  - Case-1: 200-400MB image per node
  - Case-2: 400-800MB image per node
  - Case-3: 800-1000MB image per node

Impact on Fault Management



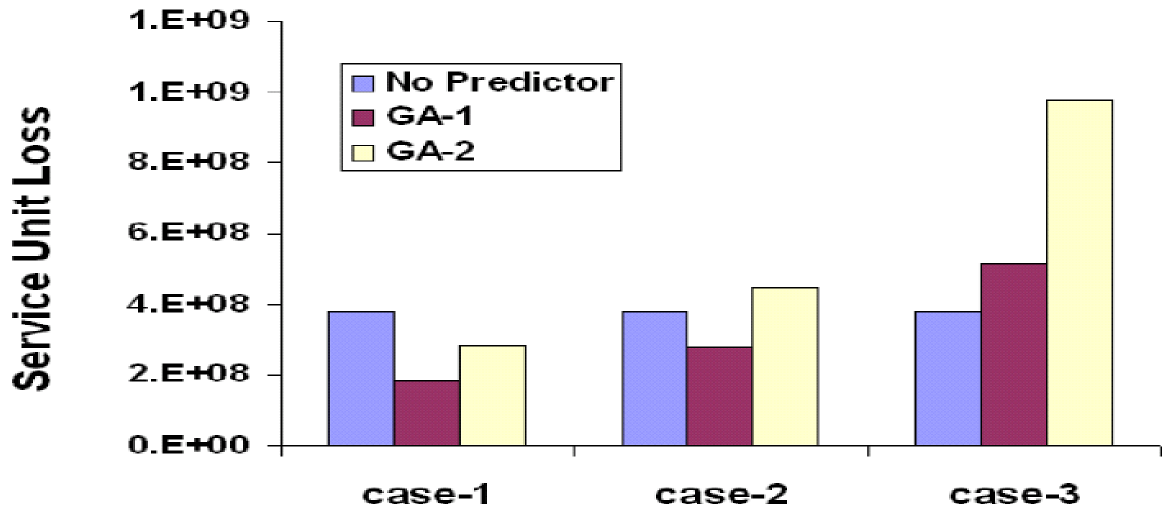
- GA-1 reduced SUL by 52.4% for case 1 (200-400MB)
  - Only 21.6% of the fatal events will actually interrupt the jobs
  - Location information is helpful to avoid meaningless checkpointing.
- GA-2 only reduced SUL by 25.1% for case 1
  - More false alarms on system level

Impact on Fault Management



- GA-1 reduced SUL by 26.6% for case-2 (400-800MB)
  - More checkpoint overhead than case-1
- GA-2 increased SUL by 18.6% for case -2
  - Insufficient lead time for checkpointing
  - significant overhead of system wide checkpointing

Impact on Fault Management



- Both GA-1 and GA-2 cannot help much in case 3(800-1000MB)
  - Both GA-1 and GA-2 generate rules without location information
  - Extreme high overhead from system-wide checkpointing
  - Failure prediction is not a good idea without location information.

Conclusions

- Location information and lead time are critical for failure prediction
- We have refined prediction metrics and presented a GA-based method to address these issues
- It can substantially boost prediction accuracy and reduce service unit loss

Our FT research website
(FENCE and RAPS projects):

http://www.cs.iit.edu/~zlan/projects.html

Acknowledgement:

- NSF CNS-0834514, CNS-0720549 and CCF-070737
- DOE DE-AC02-06CH11357